Question 1

~~Q1 Process (O_{11}, M_1, 0) → Process (O_{21}, M_2, 10) → Process (O_{31}, M~~
~~→ Process (O_{12}, M_2, ...)~~

Q1 Process (O_{11}, M_1, t_1)
J_1 Arrival time $= 0$
M_1 idle time is 0
$t_1 = 40$

Process (O_{21}, M_2, t_2)
J_2 Arrival time $= 10$
M_2 idle $= 0$
$t_2 = 10$

Process (O_{31}, M_1, t_3)
J_3 Arrival time $= 20$
$M_1 = t_1 + 50$
$t_3 = 50$

Process (O_{12}, M_2, t_4)
~~...~~
~~...~~
Operation finishes at $t_1 + 50 = 50$
M_2 finishes at $t_2 + 30 = 40$
$t_4 = \cancel{40}$ 50 due to O_{11} from process (O_{11}, M_1, t_1)

→ Process (O_{22}, M_1, t_5)
J_2 Arrival $= 10$
$M_1 = t_3 + 40 = 90$
$t_5 = 90$

Process (O_{32}, M_2, t_6)
$J_3 = 20$
$M_2 = t_4 + 25 = 75$
$t_6 = \cancel{t_3 + 40 = 90}$
$= \cancel{t_3 + 40 + 20 = 110}$
$= t_3 + 40 = 90$

Question 2

Q2 Job J1:

Last operation was O_{12}

Finish of $O_{12} = t_4 + 25 = 50 + 25 = 75$

Job J2:

Last op was O_{22}

Finish time of $O_{22} = t_5 + 35 = 90 + 35 = 125$

Job J3:

Last op was O_{32}

Finish of $O_{32} = t_6 + 20 = 90 + 20 = 110$

Makespan:

$\max(75 / 125 / 110)$

$= 125$

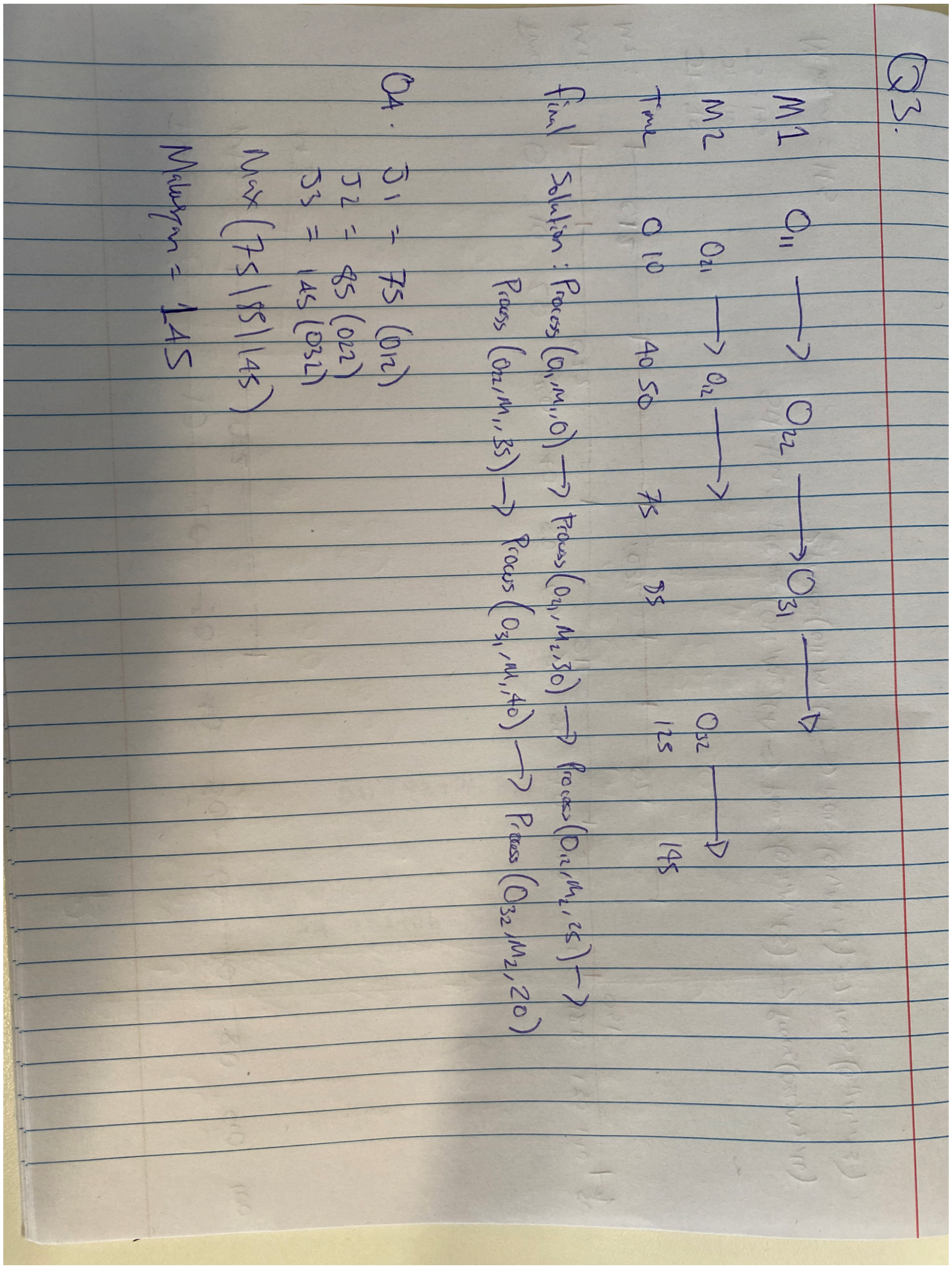
$J_1 = 75$

$J_2 = 125$

$J_3 = 110$

Makespan $= 125$

Question 3 and Question 4


Q3.

M1 $O_{11} \rightarrow O_{22} \rightarrow O_{31} \rightarrow$

M2 $O_{21} \rightarrow O_{12} \rightarrow$ $O_{32} \rightarrow$

Time 0 10 40 50 75 85 125 145

Final Solution: Process$(O_{11}, M_1, 0) \rightarrow$ Process$(O_{21}, M_2, 30) \rightarrow$ Process$(O_{12}, M_2, 25) \rightarrow$ Process$(O_{22}, M_1, 35) \rightarrow$ Process$(O_{31}, M_1, 40) \rightarrow$ Process$(O_{32}, M_2, 20)$

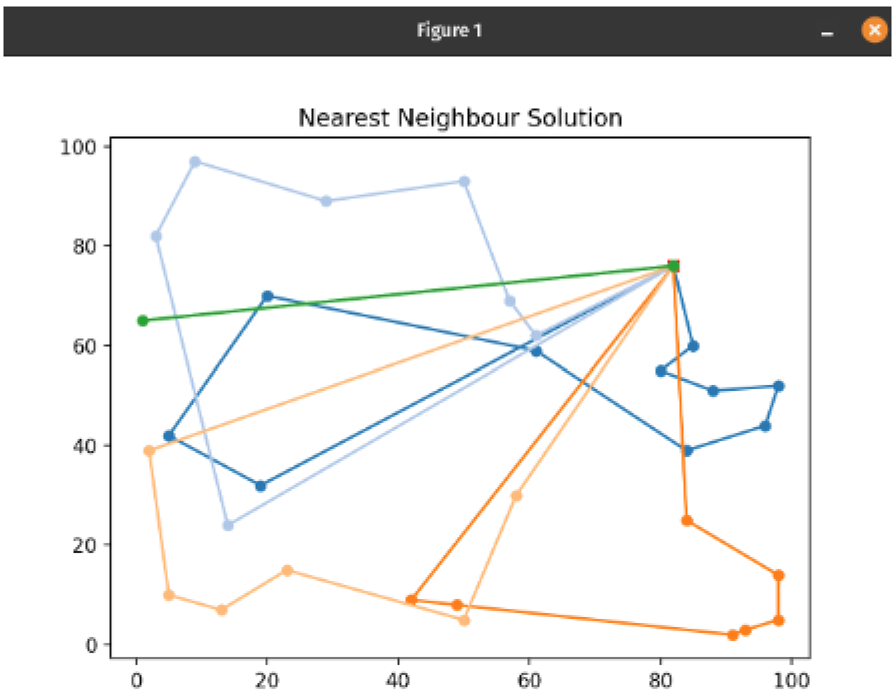
Q4. J1 = 75 (O12)
J2 = 85 (O22)
J3 = 145 (O32)
Max (75|85|145)
Makespan = 145

Using the FCFS rule the makespan came to 125 but with the SPT rule the makespan came to 145 showing a delay of 20 minutes compared to the FCFS rule.

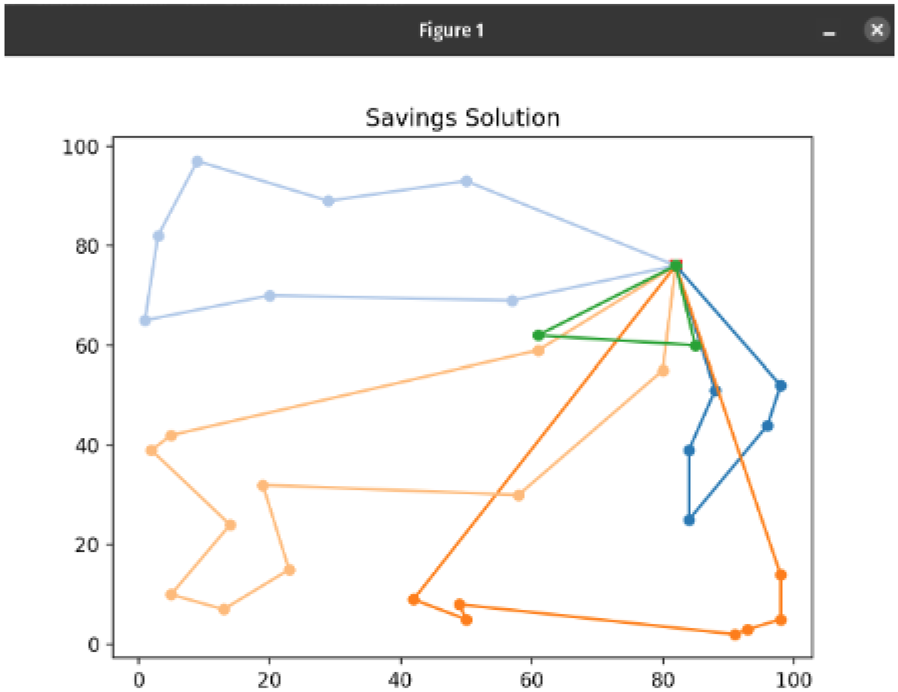
Question 5

The SPT rule is considered to be a greedy method to calculate processes. It is greedy in the sense that it always takes the shortest time that the order takes to complete, however this ignores other time aspects such as arrival, order of operations and other rule constraints that may hinder the time of actual completion. This is true in our operation as it negates the other rules surrounding the machines and job which results in a slower completion and higher makespan of 20min than the FCFS method.

Distance Discussion of Program Code

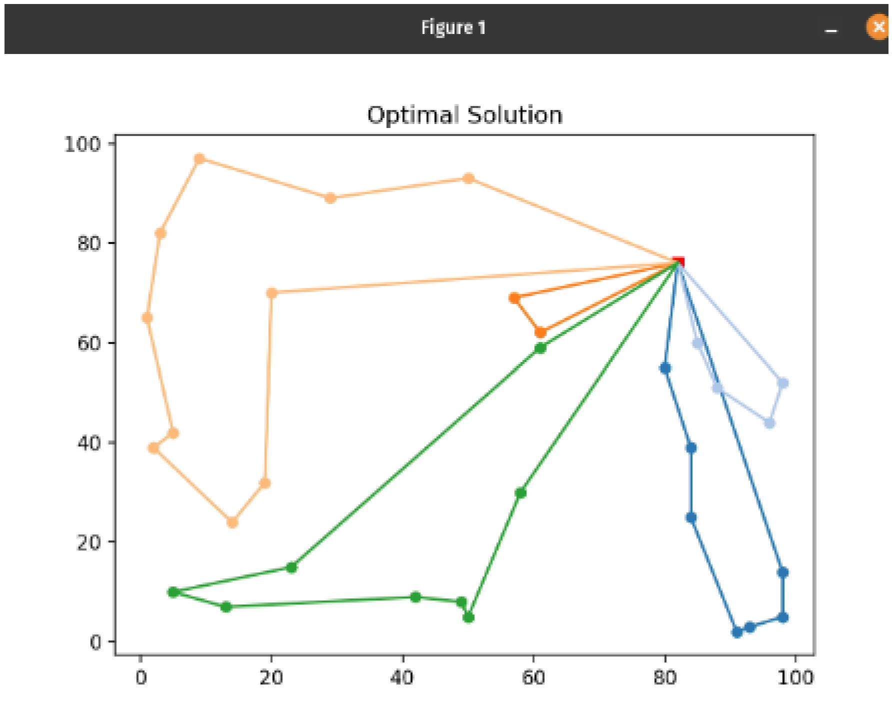


Nearest Neighbour:
Route #1: 0 30 26 16 12 1 7 14 29 22 18 0
Route #2: 0 24 27 20 5 25 10 8 0
Route #3: 0 13 21 31 19 17 3 23 0
Route #4: 0 6 2 28 4 11 9 0
Route #5: 0 15 0
Cost 1146.3996317253793



Saving Solution:
Route #1: 0 16 7 13 1 12 0
Route #2: 0 20 5 25 10 15 29 27 0
Route #3: 0 23 2 3 17 19 31 21 0
Route #4: 0 26 6 18 28 4 11 8 9 22 14 0
Route #5: 0 24 30 0
Cost 843.6881693466271

Optimal:



Route #1: 21 31 19 17 13 7 26
Route #2: 12 1 16 30
Route #3: 27 24
Route #4: 29 18 8 9 22 15 10 25 5 20
Route #5: 14 28 11 4 23 3 2 6
Cost 784

N80-k10 Nearest Neighbour
Route #1: 0 49 73 36 42 51 77 3 29 31 0
Route #2: 0 40 21 1 7 10 71 14 33 0
Route #3: 0 13 53 66 67 70 38 58 50 76 0
Route #4: 0 74 60 39 17 27 59 0
Route #5: 0 62 23 44 12 5 30 6 0
Route #6: 0 63 11 34 24 2 37 8 68 0
Route #7: 0 72 45 22 32 4 54 9 15 64 47 0
Route #8: 0 52 28 79 48 18 78 20 0
Route #9: 0 46 25 41 55 56 69 65 35 26 19 75 16 0
Route #10: 0 61 57 43 0
Cost 2255.461886842153

N80-k10 Savings

Route #1: 0 17 31 27 59 30 6 24 5 0
Route #2: 0 13 74 29 60 3 42 0
Route #3: 0 58 32 4 22 45 50 70 53 0
Route #4: 0 11 52 28 79 18 48 14 0
Route #5: 0 49 73 38 66 67 36 0
Route #6: 0 12 44 23 62 63 71 10 0
Route #7: 0 19 26 35 65 69 56 47 57 61 16 43 68 0
Route #8: 0 34 2 37 8 78 20 75 25 41 77 51 0
Route #9: 0 76 72 54 9 55 15 33 46 64 39 0
Route #10: 0 1 7 21 40 0
Cost 1836.8389976301555

N80-k10 Optimal
Route #1: 1 7 21 40
Route #2: 10 63 11 24 6 23
Route #3: 13 74 60 39 3 77 51
Route #4: 17 31 27 59 5 44 12 62
Route #5: 29 20 75 57 19 26 35 65 69 56 47 15 33 64
Route #6: 30 78 61 16 43 68 8 37 2 34
Route #7: 38 72 54 9 55 41 25 46
Route #8: 42 53 66 67 36 73 49
Route #9: 52 28 79 18 48 14 71
Route #10: 58 32 4 22 45 50 76 70
Cost 1763

N80-k10 Optimal

Differences

The Nearest Neighbour heuristic is a simple and fast approach that selects the nearest unvisited node iteratively, resulting in a total distance of 2255.46. However, it often leads to suboptimal solutions due to its focus on short-term gains without considering the overall route structure, potentially getting trapped in local optima. The Savings heuristic, with a significantly reduced total distance of 1836.84, evaluates potential savings from combining routes, aiming to minimize the total travel distance. It strikes a balance between computational efficiency and solution quality, making it a good compromise. The Optimal solution, with the lowest total distance of 1763, represents the best possible configuration of routes to minimize the total travel distance. It is obtained through complex algorithms or exhaustive search methods that consider the entire problem space, leading to globally optimal decisions. While computationally expensive, it offers the greatest efficiency and cost savings, making it ideal for high-stakes logistics and transportation planning scenarios.

From looking at the visualisation we can see that the optimal solution has routes that are quite separate, each route has its own area and does not cross into another route’s path.
When we compare this to the nearest neighbour solution, we see that the nearest neighbours sometimes causes a route which crosses multiple other routes, which is not very efficient, as those nodes could be collected by a route already in the area.
The savings heuristic does a better job at this, due to the nature of computing the largest savings, but still has some routes that cross each other